

```

EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTT

```

EXE

Mod

ED

ED

ED

ED

ED

ED

ED

ED

ED 1

ED

SYS
LIB

LI

```
RRRRRRRR      AAAAAA      NN      NN      NN      NN      EEEEEEEEEEE      XX      XX      TTTTTTTTTT
RRRRRRRR      AAAAAA      NN      NN      NN      NN      EEEEEEEEEEE      XX      XX      TTTTTTTTTT
RR      RR      AA      AA      NN      NN      NN      NN      EE      XX      XX      TT
RR      RR      AA      AA      NN      NN      NN      NN      EE      XX      XX      TT
RR      RR      AA      AA      NNNN      NN      NNNN      NN      EE      XX      XX      TT
RR      RR      AA      AA      NNNN      NN      NNNN      NN      EE      XX      XX      TT
RRRRRRRR      AA      AA      NN      NN      NN      NN      EEEEEEEEEEE      XX      XX      TT
RRRRRRRR      AA      AA      NN      NN      NN      NN      EEEEEEEEEEE      XX      XX      TT
RR      RR      AAAAAAAAAA      NN      NNNN      NN      NNNN      EE      XX      XX      TT
RR      RR      AAAAAAAAAA      NN      NNNN      NN      NNNN      EE      XX      XX      TT
RR      RR      AA      AA      NN      NN      NN      NN      EE      XX      XX      TT
RR      RR      AA      AA      NN      NN      NN      NN      EE      XX      XX      TT
RR      RR      AA      AA      NN      NN      NN      NN      EEEEEEEEEEE      XX      XX      TT
RR      RR      AA      AA      NN      NN      NN      NN      EEEEEEEEEEE      XX      XX      TT
```

```
LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```



```
0001 0 %TITLE 'EDT$RANNEXT - next line in a range'
0002 0 MODULE EDT$RANNEXT (
0003 0 IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 Get the next line in a range.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: February 3, 1978
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 19-FEB-1981. This module was created by
0045 1 extracting routine EDT$NXT_LNRNG from module RANGE.
0046 1 1-003 - Use the ASSERT macro. JBS 01-Jun-1981
0047 1 1-004 - If this is an ALL range, it must have a search string.
0048 1 JBS 31-Oct-1981
0049 1 1-005 - Use the new PREV_RANGE field to find the ALL string. JBS 02-Nov-1981
0050 1 1-006 - Modify the next range find for AND ranges. SMB 15-Feb-1982
0051 1 1-007 - Type .,- now works at the [EOB]. TMV 18-FEB-1982
0052 1 1-008 - Regularize format. JBS 04-May-1982
0053 1 1-009 - Remove EDT$A_STR_CMP. JBS 16-Jul-1982
0054 1 1-010 - Put calls to edt$rng_nxtln in line. STS 17-Sep-1982
0055 1 1-011 - Put edt$stst_eob in line. STS 22-Sep-1982
0056 1 1-012 - Remove reference to edt$rng_posfrst. STS 11-Oct-1982
0057 1 1-013 - Modify to use new cmpare macro. STS 20-Oct-1982
```

EDTSRANNEXT
V04-000

EDTSRANNEXT - next line in a range

M 7
16-Sep-1984 01:25:17
14-Sep-1984 12:24:16

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]RANNEXT.BLI;1 Page (1) 2

: 58
: 59
: 60

0058 1 ! 1-014 - Improve the appearance of the listing. JBS 17-Jun-1983
0059 1 !--
0060 1

EDTSRANNEXT
V04-000

EDTSRANNEXT - next line in a range
Declarations

N 7
16-Sep-1984 01:25:17
14-Sep-1984 12:24:16

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]RANNEXT.BLI;1
Page 3
(2)

```
: 62      0061 1 %SBTTL 'Declarations'
: 63      0062 1
: 64      0063 1 | TABLE OF CONTENTS:
: 65      0064 1 |
: 66      0065 1 |
: 67      0066 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 68      0505 1
: 69      0506 1 FORWARD ROUTINE
: 70      0507 1     EDT$NXT_LNRNG;
: 71      0508 1
: 72      0509 1 |
: 73      0510 1 | INCLUDE FILES:
: 74      0511 1 |
: 75      0512 1 |
: 76      0513 1 REQUIRE 'EDTSRC:EDTREQ';
: 77      0648 1
: 78      0649 1 |
: 79      0650 1 | MACROS:
: 80      0651 1 |
: 81      0652 1 |     NONE
: 82      0653 1 |
: 83      0654 1 | EQUATED SYMBOLS:
: 84      0655 1 |
: 85      0656 1 |     NONE
: 86      0657 1 |
: 87      0658 1 | OWN STORAGE:
: 88      0659 1 |
: 89      0660 1 |     NONE
: 90      0661 1 |
: 91      0662 1 | EXTERNAL REFERENCES:
: 92      0663 1 |
: 93      0664 1 |     In the routine
```

EDT
V04


```

95 0665 1 %SBTTL 'EDT$$NXT_LNRNG - next line in a range'
96 0666 1
97 0667 1 GLOBAL ROUTINE EDT$$NXT_LNRNG (           ! Get the next line in a range
98 0668 1   DELETED                                ! 1 = last line was deleted
99 0669 1   ) =
100 0670 1
101 0671 1 ++
102 0672 1 FUNCTIONAL DESCRIPTION:
103 0673 1
104 0674 1     This routine returns the next line in a range. The position
105 0675 1     routine must have been called first if this routine is to
106 0676 1     work properly.
107 0677 1
108 0678 1 FORMAL PARAMETERS:
109 0679 1
110 0680 1   DELETED                                Indicate whether the last line was deleted.
111 0681 1
112 0682 1 IMPLICIT INPUTS:
113 0683 1
114 0684 1     EDT$$Z_RNG_CURRNG
115 0685 1     EDT$$L_RNG_EOL
116 0686 1     EDT$$G_RNG_MORELN
117 0687 1     EDT$$G_RNG_NOOFLN
118 0688 1     EDT$$G_SEA_STRLN
119 0689 1     EDT$$A_WK_CN
120 0690 1     EDT$$G_EXCT_MATCH
121 0691 1
122 0692 1 IMPLICIT OUTPUTS:
123 0693 1
124 0694 1     EDT$$G_RNG_FRSTLN
125 0695 1
126 0696 1 ROUTINE VALUE:
127 0697 1
128 0698 1     0 = No more lines exist in this range
129 0699 1     1 = A line was sucessfully located
130 0700 1
131 0701 1
132 0702 1 SIDE EFFECTS:
133 0703 1
134 0704 1     The current text buffer is re-positioned.
135 0705 1
136 0706 1 --
137 0707 1
138 0708 1 BEGIN
139 0709 2
140 0710 2 EXTERNAL ROUTINE
141 0711 2   EDT$$RNG_REPOS,
142 0712 2   EDT$$RD_NXTLN,
143 0713 2   EDT$$STR_CMP;           ! Compare two strings of equal length
144 0714 2
145 0715 2 EXTERNAL
146 0716 2   EDT$$Z_RNG_CURRNG : REF NODE_BLOCK,      ! The current range node
147 0717 2   EDT$$L_RNG_EOL : LN_BLOCK,             ! The line number at which this range ends
148 0718 2   EDT$$G_RNG_FRSTLN,
149 0719 2   EDT$$G_RNG_MORELN,                      ! Used by EDT$$NXT_LNRNG to indicate more lines.
150 0720 2   EDT$$G_RNG_NOOFLN,                      ! Count of number of lines in a range.
151 0721 2   EDT$$G_SEA_STRLN,                      ! Length of search string.
```



```
152 0722 2 EDT$$Z_EOB_LN,  
153 0723 2 EDT$$A_WK_CN : REF LIN_BLOCK, ! The current line pointer,  
154 0724 2 EDT$$G_EXCT_MATCH; ! The type of string matching  
155 0725 2  
156 0726 2 LABEL  
157 0727 2 LOOP;  
158 0728 2  
159 0729 2 LOCAL  
160 0730 2 DEL_FLAG;  
161 0731 2  
162 0732 2 !+  
163 0733 2 Get a local copy of the delete flag, since if this is an ALL range,  
164 0734 2 we will loop, and the delete flag will be turned off after the first  
165 0735 2 time though.  
166 0736 2 -  
167 0737 2 DEL_FLAG = .DELETED;  
168 0738 2 !+  
169 0739 2 Loop until we find a line which contains the ALL string. If no all  
170 0740 2 string, we will get out the first time through.  
171 0741 2 -  
172 0742 2 LOOP :  
173 0743 3 BEGIN  
174 0744 3  
175 0745 3 WHILE 1 DO  
176 0746 4 BEGIN  
177 0747 4 !+  
178 0748 4 If we are at the end of the buffer, or the EDT$$G_RNG_MORELN flag is zero,  
179 0749 4 then there are no more lines. Exit with a 0.  
180 0750 4 -  
181 0751 4  
182 0752 4 IF (.EDT$$G_RNG_MORELN EQL 0) THEN RETURN (0);  
183 0753 4  
184 0754 4 !+  
185 0755 4 Case on the range type.  
186 0756 4 -  
187 0757 4  
188 0758 4 CASE .EDT$$Z_RNG_CURRNG [RAN_TYPE] FROM RAN_NULL TO NUM_RAN OF  
189 0759 4 SET  
190 0760 4 !+  
191 0761 4 For all the single line ranges, if EDT$$G_RNG_FRSTLN is on, then return  
192 0762 4 success and turn it off. If it is not then check to see if it  
193 0763 4 was part of an AND list; if so, position to the next line in  
194 0764 4 the list.  
195 0765 4 -  
196 0766 4  
197 0767 4 [RAN_NUMBER, RAN_DOT, RAN_STR, RAN_BEGIN, RAN_LAST, RAN_END, RAN_MINUS, RAN_PLUS, RAN_NULL,  
198 0768 4 RAN_MINSTR, RAN_ORIG]:  
199 0769 5 BEGIN  
200 0770 5  
201 0771 5 IF .EDT$$G_RNG_FRSTLN !  
202 0772 5 THEN  
203 0773 5 EDT$$G_RNG_FRSTLN = 0  
204 0774 5 ELSE  
205 0775 5  
206 0776 6 IF (.EDT$$Z_RNG_CURRNG [NEXT_RANGE] EQL 0)  
207 0777 5 THEN  
208 0778 6 RETURN (0)
```



```
209 0779 5 ELSE
210 0780 5
211 0781 5 IF EDT$SRNG_REPOS (.EDT$$Z_RNG_CURRNG [NEXT_RANGE])
212 0782 5 THEN
213 0783 5 EDT$$G_RNG_FRSTLN = 0
214 0784 5 ELSE
215 0785 5 RETURN (0);
216 0786 5
217 0787 4 END;
218 0788 4 !+
219 0789 4 !- For WHOLE or REST, just move to the next line.
220 0790 4
221 0791 4
222 0792 4 [RAN_WHOLE, RAN_REST] :
223 0793 5 BEGIN
224 0794 5
225 0795 5 IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN) THEN RETURN (0);
226 0796 5
227 0797 6 IF ( NOT .DEL_FLAG)
228 0798 5 THEN
229 0799 5
230 0800 5 IF ( NOT .EDT$$G_RNG_FRSTLN) THEN EDT$SRD_NXTLN () ELSE EDT$$G_RNG_FRSTLN = 0;
231 0801 5
232 0802 4 END;
233 0803 4 !+
234 0804 4 !- For FOR and SELECT, the range block contains the number of lines
235 0805 4 to include. Count it down, returning failure if it becomes negative.
236 0806 4
237 0807 4
238 0808 4 [RAN_FOR, RAN_SELECT] :
239 0809 5 BEGIN
240 0810 5
241 0811 5 IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN) THEN RETURN (0);
242 0812 5
243 0813 5 EDT$$Z_RNG_CURRNG [RAN_VAL] = .EDT$$Z_RNG_CURRNG [RAN_VAL] - 1;
244 0814 5
245 0815 6 IF (.EDT$$Z_RNG_CURRNG [RAN_VAL] LSS 0)
246 0816 5 THEN
247 0817 6 RETURN (0)
248 0818 5 ELSE
249 0819 5
250 0820 6 IF ( NOT .DEL_FLAG)
251 0821 5 THEN
252 0822 5
253 0823 5 IF ( NOT .EDT$$G_RNG_FRSTLN) THEN EDT$SRD_NXTLN () ELSE EDT$$G_RNG_FRSTLN = 0;
254 0824 5
255 0825 4 END;
256 0826 4 !+
257 0827 4 !- The THRU and BEFORE ranges have saved away the last line to be
258 0828 4 included. Move to the next line in the range, then compare it
259 0829 4 to the last line number. If it is greater, return failure.
260 0830 4
261 0831 4
262 0832 4 [RAN_THRU, RAN_BEFORE] :
263 0833 5 BEGIN
264 0834 5
265 0835 5 IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN) THEN RETURN (0);
```



```
266 0836 5
267 0837 6      IF ( NOT .DEL_FLAG)
268 0838 5      THEN
269 0839 5          IF ( NOT .EDT$$G_RNG_FRSTLN) THEN EDT$$RD_NXTLN ( ) ELSE EDT$$G_RNG_FRSTLN = 0;
270 0840 5          IF (CMPLNO (EDT$$A_WK_LN [LIN_NUM], EDT$$L_RNG_EOL) GTR 0) THEN RETURN (0);
271 0841 5
272 0842 5      END;
273 0843 4
274 0844 4      [INRANGE] :
275 0845 4      ;
276 0846 4      ;
277 0847 4      ;
278 0848 4      ;
279 0849 4      [OUTRANGE] :
280 0850 4      ASSERT (0);
281 0851 4      TES;
282 0852 4
283 0853 4
284 0854 4      !+ If there was an ALL, insure that the string exists in the line.
285 0855 4      ! If it does not, continue through the loop again looking either
286 0856 4      ! for one that does contain the string or the end of the range.
287 0857 4      !-
288 0858 5      BEGIN
289 0859 5
290 0860 5      BIND
291 0861 5          ALL_RAN = .EDT$$Z_RNG_CURRNG [PREV_RANGE] : NODE_BLOCK;
292 0862 5
293 0863 5      LOCAL
294 0864 5          CH_POINT;
295 0865 5
296 0866 5      IF (ALL_RAN EQLA 0) THEN LEAVE LOOP;
297 0867 5
298 0868 5      IF ((ALL_RAN NEQA 0) AND (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN)) THEN RETURN (0);
299 0869 5
300 0870 5      IF (.ALL_RAN [RAN_TYPE] NEQ RAN_ALL) THEN LEAVE LOOP;
301 0871 5
302 0872 5      !+
303 0873 5      ! An ALL range must have a string.
304 0874 5      !-
305 0875 5      ASSERT (.ALL_RAN [STR_PNT] NEQA 0);
306 0876 5      CH_POINT = CH$PTR (EDT$$A_WK_LN [LIN_TEXT]);
307 0877 5
308 0878 5      DECR I FROM .EDT$$A_WK_LN [LIN_LENGTH] - .EDT$$G_SEA_STRLN TO 0 DO
309 0879 5
310 0880 5          IF EDT$$STR_CMP (.CH_POINT, .ALL_RAN [STR_PNT], .ALL_RAN [RAN_VAL], .EDT$$G_EXCT_MATCH)
311 0881 5          THEN
312 0882 5              LEAVE LOOP
313 0883 5              ! This line has the string
314 0884 5          ELSE
315 0885 5              CH_POINT = CH$PLUS (.CH_POINT, 1);
316 0886 5              ! Keep looking
317 0887 5
318 0888 5      DEL_FLAG = 0;
319 0889 5      END;
320 0890 5      END;
321 0891 5      !+
322 0892 5      ! Count one more line found in the range.
```

EDTSRANNEXT
V04-000

EDTSRANNEXT - next line in a range
EDT\$\$NXT_LNRNG - next line in a range

F 8
16-Sep-1984 01:25:17
14-Sep-1984 12:24:16

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]RANNEXT.BLI;1
Page 8
(3)

```
: 323      0893  2  !-
: 324      0894  2      EDT$$G_RNG_NOOFLN = .EDT$$G_RNG_NOOFLN + 1;
: 325      0895  2      RETURN(1);
: 326      0896  1      END;
```

! of routine EDT\$\$NXT_LNRNG

.TITLE EDTSRANNEXT EDTSRANNEXT - next line in a range
.IDENT \V04-000\

.EXTRN EDT\$\$RNG_REPOS, EDT\$\$RD NXTLN
.EXTRN EDT\$\$STR_CMP, EDT\$\$Z RNG_CURRNG
.EXTRN EDT\$\$L RNG_EOL, EDT\$\$G_RNG_FRSTLN
.EXTRN EDT\$\$G_RNG_MORELN
.EXTRN EDT\$\$G_RNG_NOOFLN
.EXTRN EDT\$\$G_SEA_STRLEN
.EXTRN EDT\$\$Z_EOB_LN, EDT\$\$A_WK_LN
.EXTRN EDT\$\$G_EXCT_MATCH
.EXTRN EDT\$\$INTER_ERR

.PSECT _EDT\$CODE, NOWRT, SHR, PIC, 2

.ENTRY EDT\$\$NXT_LNRNG, Save R2,R3,R4,R5,R6,R7,R8,- : 0667
R9,R10,RT1

MOVAB EDT\$\$RD NXTLN, R11
MOVAB EDT\$\$INTER_ERR, R10
MOVAB EDT\$\$Z_RNG_CURRNG, R9
MOVAB EDT\$\$Z_EOB_LN, R8
MOVAB EDT\$\$G_RNG_FRSTLN, R7
MOVAB EDT\$\$A_WK_LN, R6
MOVL DELETED, DEL_FLAG
TSTL EDT\$\$G_RNG_MORELN
BEQL 4\$
MOVL EDT\$\$Z_RNG_CURRNG, R2
CASEB 1(R2), #0, #20
.WORD 3\$-2\$,-
3\$-2\$,-
3\$-2\$,-
3\$-2\$,-
3\$-2\$,-
3\$-2\$,-
3\$-2\$,-
17\$-2\$,-
3\$-2\$,-
10\$-2\$,-
5\$-2\$,-
5\$-2\$,-
6\$-2\$,-
17\$-2\$,-
3\$-2\$,-
3\$-2\$,-
6\$-2\$,-
10\$-2\$,-
3\$-2\$,-
17\$-2\$,-
17\$-2\$,-

CALLS #0, EDT\$\$INTER_ERR : 0850
BRB 9\$: 0758

OFFC 00000

```
5B 00000000G 00 9E 00002  
5A 00000000G 00 9E 00009  
59 00000000G 00 9E 00010  
58 00000000G 00 9E 00017  
57 00000000G 00 9E 0001E  
56 00000000G 00 9E 00025  
55 00000000G 00 9E 0002C  
04 AC D0 0002C 1$:  
00000000G 00 D5 00030  
4C 13 00036  
52 69 D0 00038  
00 01 A2 8F 0003B 2$:  
002F 002F 002F 002F 00040  
00B1 002F 002F 002F 00048  
0047 006D 002F 002F 00050  
002F 00B1 0051 00058  
00B1 006D 0051 00060  
00B1 00068
```

```
002F      14  
00B1      002F  
0047      002F  
002F      0047  
00B1      002F
```

```
002F      002F  
002F      002F  
006D      002F  
00B1      002F  
006D      002F
```

```
6A      00 FB 0006A  
3C      11 0006D
```


37	67	E8	0006F	3\$:	BLBS	EDT\$\$G_RNG_FRSTLN, 8\$	0771
	A2	D5	00072		TSTL	16(R2)	0776
	0D	13	00075		BEQL	4\$	
	A2	DD	00077		PUSHL	16(R2)	0781
00000000G	00	01	FB	0007A	CALLS	#1, EDT\$\$RNG_REPOS	
25	50	E8	00081		BLBS	R0, 8\$	
	00CA	31	00084	4\$:	BRW	22\$	
50	68	9E	00087	5\$:	MOVAB	EDT\$\$Z_EOB_LN, R0	0795
50	66	D1	0008A		CMPL	EDT\$\$A_WK_LN, R0	
	F5	13	0008D		BEQL	4\$	
	0D	11	0008F		BRB	7\$	0797
50	68	9E	00091	6\$:	MOVAB	EDT\$\$Z_EOB_LN, R0	0811
50	66	D1	00094		CMPL	EDT\$\$A_WK_LN, R0	
	EB	13	00097		BEQL	4\$	
	04	A2	D7	00099	DECL	4(R2)	0813
	E6	19	0009C		BLSS	4\$	0815
50	55	E8	0009E	7\$:	BLBS	DEL FLAG, 17\$	0820
05	67	E8	000A1		BLBS	EDT\$\$G_RNG_FRSTLN, 8\$	0823
6B	00	FB	000A4		CALLS	#0, EDT\$\$RD_NXTLN	
	48	11	000A7		BRB	17\$	
	67	D4	000A9	8\$:	CLRL	EDT\$\$G_RNG_FRSTLN	
	44	11	000AB	9\$:	BRB	17\$	
50	68	9E	000AD	10\$:	MOVAB	EDT\$\$Z_EOB_LN, R0	0835
50	66	D1	000B0		CMPL	EDT\$\$A_WK_LN, R0	
	CF	13	000B3		BEQL	4\$	
0A	55	E8	000B5		BLBS	DEL FLAG, 12\$	0837
05	67	E8	000B8		BLBS	EDT\$\$G_RNG_FRSTLN, 11\$	0840
6B	00	FB	000BB		CALLS	#0, EDT\$\$RD_NXTLN	
	02	11	000BE		BRB	12\$	
	67	D4	000C0	11\$:	CLRL	EDT\$\$G_RNG_FRSTLN	
50	66	01	C1	000C2	ADDL3	#1, EDT\$\$A_WK_LN, R0	0842
51	00000000G	00	3C	000C6	MOVZWL	HIGH 2, R1	
51	04	A0	B1	000CD	CMPL	4(R0), R1	
	0E	1F	000D1		BLSSU	13\$	
	17	12	000D3		BNEQ	15\$	
51	00000000G	00	D0	000D5	MOVL	LOW 2, R1	
51		60	D1	000DC	CMPL	(R0), R1	
	05	1E	000DF		BGEQU	14\$	
50	01	CE	000E1	13\$:	MNEGL	#1, R0	
	09	11	000E4		BRB	16\$	
	04	12	000E6	14\$:	BNEQ	15\$	
	50	D4	000E8		CLRL	R0	
	03	11	000EA		BRB	16\$	
50	01	D0	000EC	15\$:	MOVL	#1, R0	
	60	14	000EF	16\$:	BGTR	22\$	
50	69	D0	000F1	17\$:	MOVL	EDT\$\$Z_RNG_CURRNG, R0	0861
53	14	A0	D0	000F4	MOVL	20(R0), R3	
	4D	13	000F8		BEQL	21\$	0866
50	68	9E	000FA		MOVAB	EDT\$\$Z_EOB_LN, R0	0868
50	66	D1	000FD		CMPL	EDT\$\$A_WK_LN, R0	
	4F	13	00100		BEQL	22\$	
13	01	A3	91	00102	CMPB	1(R3), #19	0870
	3F	12	00106		BNEQ	21\$	
	08	A3	D5	00108	TSTL	8(R3)	0875
	03	12	0010B		BNEQ	18\$	
6A	00	FB	0010D		CALLS	#0, EDT\$\$INTER_ERR	
50	66	D0	00110	18\$:	MOVL	EDT\$\$A_WK_LN, R0	0876

EDT\$RANNEXT
V04-000

EDT\$RANNEXT - next line in a range
EDT\$\$NXT_LNRNG - next line in a range

H 8
16-Sep-1984 01:25:17
14-Sep-1984 12:24:16

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]RANNEXT.BLI;1

Page 10
(3)

52	07	A0	9E	00113	MOVAB	7(R0), CH_POINT	:	
54		60	9A	00117	MOVZBL	(R0), R4	:	0878
54	00000000G	00	C2	0011A	SUBL2	EDT\$\$G_SEA_STRLEN, R4	:	
		54	D6	00121	INCL	I	:	
		1A	11	00123	BRB	20\$	:	
	00000000G	00	DD	00125	PUSHL	EDT\$\$G_EXCT_MATCH	:	0880
	04	A3	DD	0012B	PUSHL	4(R3)	:	
	08	A3	DD	0012E	PUSHL	8(R3)	:	
		52	DD	00131	PUSHL	CH_POINT	:	
00000000G	00	04	FB	00133	CALLS	#4, EDT\$\$STR_CMP	:	
	0A	50	E8	0013A	BLBS	R0, 21\$	:	
		52	D6	0013D	INCL	CH_POINT	:	0884
	E3	54	F4	0013F	SOBGEQ	I, 19\$	:	0880
		55	D4	00142	CLRL	DEL_FLAG	:	0836
		FEE9	31	00144	BRW	1\$	:	0745
	00000000G	00	D6	00147	INCL	EDT\$\$G_RNG_NOOFLN	:	0894
	50	01	D0	0014D	MOVL	#1, R0	:	0895
			04	00150	RET		:	
		50	D4	00151	CLRL	R0	:	0896
			04	00153	RET		:	

; Routine Size: 340 bytes, Routine Base: _EDT\$CODE + 0000

; 327 0897 1
; 328 0898 1 !<BLF/PAGE>

EDT\$RANNEXT
V04-000

EDT\$RANNEXT - next line in a range
EDT\$\$NXT_LNRNG - next line in a range

I 8
16-Sep-1984 01:25:17
14-Sep-1984 12:24:16

VAX-11 Bliss-32 V4.0-742
DISK\$VM\$MASTER:[EDT.SRC]RANNEXT.BLI;1 Page 11 (4)

: 330 0899 1 END
: 331 0900 1
: 332 0901 0 ELUDOM

! of module EDT\$RANNEXT

PSECT SUMMARY

: Name Bytes Attributes
: _EDT\$CODE 340 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	83	22	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:RANNEXT/OBJ=OBJ\$:RANNEXT MSRC\$:RANNEXT.BLI/UPDATE=(ENH\$:RANNEXT
:)

: Size: 340 code + 0 data bytes
: Run Time: 00:20.9
: Elapsed Time: 00:26.7
: Lines/CPU Min: 2587
: Lexemes/CPU-Min: 9518
: Memory Used: 151 pages
: Compilation Complete

0138

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY